SAP NetWeaver 7.0 (2004s)

Configuration Guide

SAP Interactive Forms by Adobe

Adobe Document Services

For SAP NetWeaver 7.0

(Support Package SPS 13)

Document Version 1.3 – September 2007

THE BEST-RUN BUSINESSES RUN SAP

THE BEST-RUN BUSINESSES RUN SAP

SAP AG
Dietmar-Hopp-Allee 16
69190 Walldorf
Germany
T +49/18 05/34 34 34
F +49/18 05/34 34 20
www.sap.com

Copyright

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressively prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Documentation in the SAP Service Marketplace

You can find this documentation at the following Internet address:
`service.sap.com/adobe` → *Media Library* → *Documentation.*

Icons in Body Text

Icon	Meaning
⚠	Caution
❖	Example
💡	Note
⬆	Recommendation
<>	Syntax

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see *Help on Help → General Information Classes and Information Classes for Business Information Warehouse* on the first page of any version of *SAP Library*.

Typographic Conventions

Type Style	Description
Example text	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation.
`Example text`	Emphasized words or phrases in body text, graphic titles, and table titles.
EXAMPLE TEXT	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
`Example text`	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example text`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, `F2` or `ENTER`.

1 Adobe Document Services Configuration Guide

Introduction

Adobe® document services enhance the document handling capabilities of SAP Web Application Server (SAP Web AS). Adobe document services allow SAP applications (either Java or ABAP) to take advantage of the full range of capabilities in Adobe Acrobat® Professional, Adobe Acrobat Standard, and Adobe Reader®. These capabilities enable SAP customers to:

- Create and deploy interactive forms that look exactly like their paper counterparts
- Work with forms in online and offline scenarios
- Annotate PDF documents and collaborate on PDF document reviews
- Generate dynamic PDF documents from data contained in the SAP system
- Capture data using forms and import that data directly into the SAP system
- Allow users to digitally sign PDF documents
- Embed other file formats inside PDF documents as attachments

Target Groups

This guide describes how to configure Adobe document services. It is aimed at the system administrator and assumes familiarity with the SAP Web AS installation and configuration.

Related Documentation

The programmatic interface to Adobe document services is described in the documentation for the PDF object. There is a PDF object interface for both ABAP and Java environments. Both interfaces provide the same functionality, but they each expose it in an object-oriented manner appropriate to the programming language they serve.

To develop the form designs for use with Adobe document services, the form author uses Adobe LiveCycle Designer, which is accessible from a number of environments in SAP including:

- SAP NetWeaver Developer Studio in the Web Dynpro section
- ABAP Workbench in the Form Builder section (transaction SFP)

For information on how to develop form designs, see the documentation provided with the Adobe LiveCycle Designer installation.

> Check the newest printable version of this Configuration Guide in the SAP Service Marketplace, available at **`http://service.sap.com/adobe`** → *Media Library* → *Documentation*.

2 Important SAP Notes

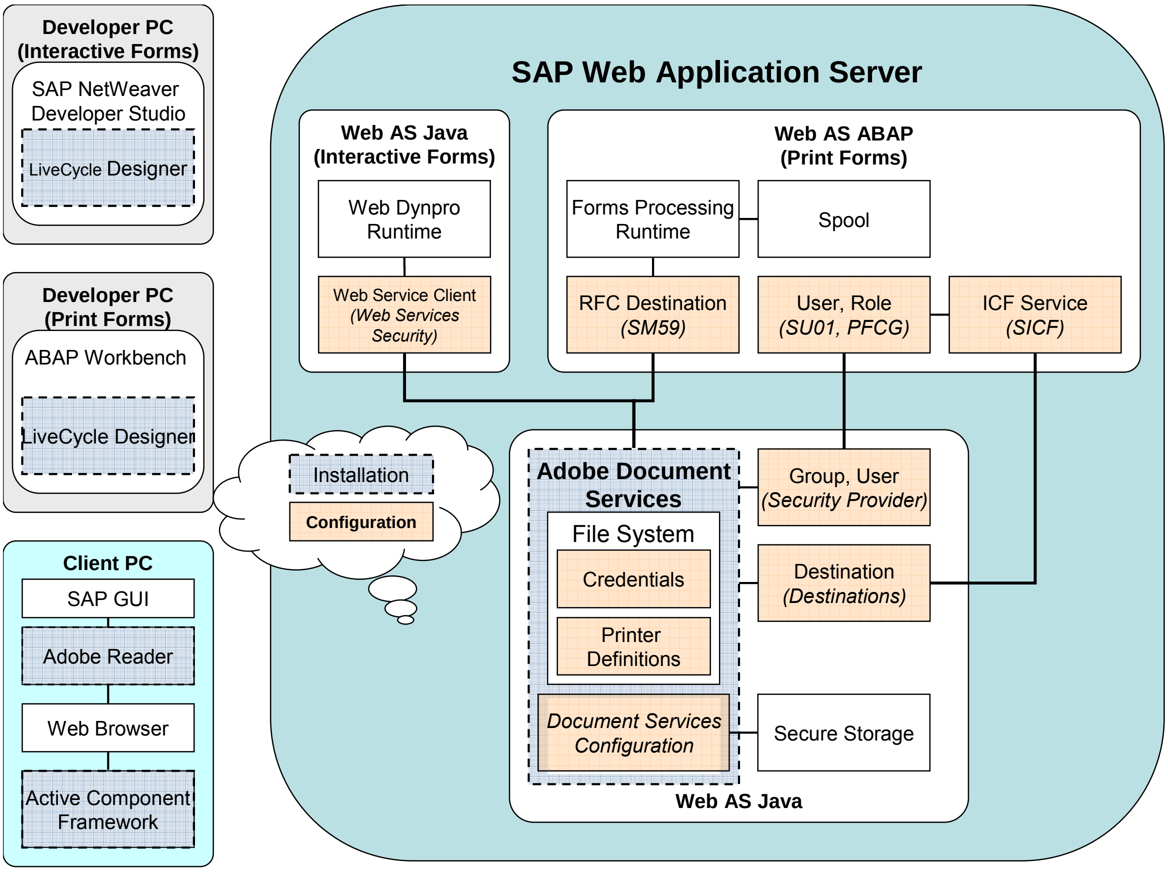
The most important SAP Notes that apply to the configuration of the Adobe document services are shown in the table below.

Important SAP Notes

SAP Note Number	Title
894009	Adobe document services: Configuration Guide (NW 7.0)
736902	Adobe Credentials
750784	SAP Interactive Forms: Licenses
752153	Adobe: PDF Manipulation Module High Encryption
766191	Active Component Framework Installation
766410	Interactive forms: XDC Scenarios for the Printer Control
685571	Printing PDF-Based Forms
834573	SAP Interactive Forms by Adobe: Acrobat/Adobe Reader Version
854621	Third Party Software Delivered with Adobe Document Services and Adobe LiveCycle Designer
944221	Problem Analysis / Troubleshooting in Forms Processing
886572	ADS: Central Patch Note for SAP NetWeaver 7.0

3 Architecture

The figure below gives you an overview of the architecture of SAP Interactive Forms by Adobe. It shows the parts that have to be installed and the parts that you have to configure as described in this documentation. You can also see the communication paths between the components used in SAP Web AS.



LiveCycle Designer is installed locally on the developer's PC and integrated into the following development environments:

- ABAP Workbench
- SAP NetWeaver Developer Studio

You install LiveCycle Designer from the separate CD/DVD delivered by SAP.

The installation of Adobe Reader and the Active Component Framework is described in the section Additional Installations on the Client PC [Page 51].

The configuration steps for SAP Interactive Forms in SAP Web AS are described in the following chapters under *Configuring the Web Service*. The figure shows you at a glance which transactions in the ABAP system or service nodes in the Visual Administrator in the Java System you need to use.

4 Checking the IIOP Service and the Startup Properties

Use

After the installation of the Adobe document services and before performing any other configuration steps you need to check special settings on the central instance host. If necessary, you need to adapt these settings manually.

Procedure

1. Start the configuration tool of the AS Java:

− On Windows:

Run **`/usr/sap/<SAPSID>/<instance_name>/j2ee/configtool.bat`**

− On Linux:

Run **`/usr/sap/<SAPSID>/<instance_name>/j2ee/configtool.sh`**

The Config Tool screen appears.

2. Check if the startup mode of the service iiop is set to always:
   a. In the left frame, open the tree *Cluster data* → *Global dispatcher configuration* → *Services*.
   b. Choose service *iiop*.
   c. The field *Startup mode* in the right frame must be set to always. If it is not true, apply the value always.
   d. Choose *Apply changes*.
   e. Repeat the steps b. to d. for the following path: *Cluster data* → *Global server configuration* → *Services.*
3. Check if the Java startup property for Adobe document services is set:
   a. Select *Cluster data* → *instance_<IDxxxx>* → *server_<IDxxxx>.*
   b. In the right frame, check if the following line exists in the Java Parameters area of the tab General:

      -Dorg.omg.PortableInterceptor.ORBInitializerClass.com.sap.engine.services.ts.jts.ots.PortableInterceptor.JTSInitializer

      If the line does not exist, add the line to this section.
4. Exit the configuration tool.
5. If you have applied new values during the procedure above, you need to restart the AS Java to adapt the new settings.

5 Configuring the Web Service

Purpose

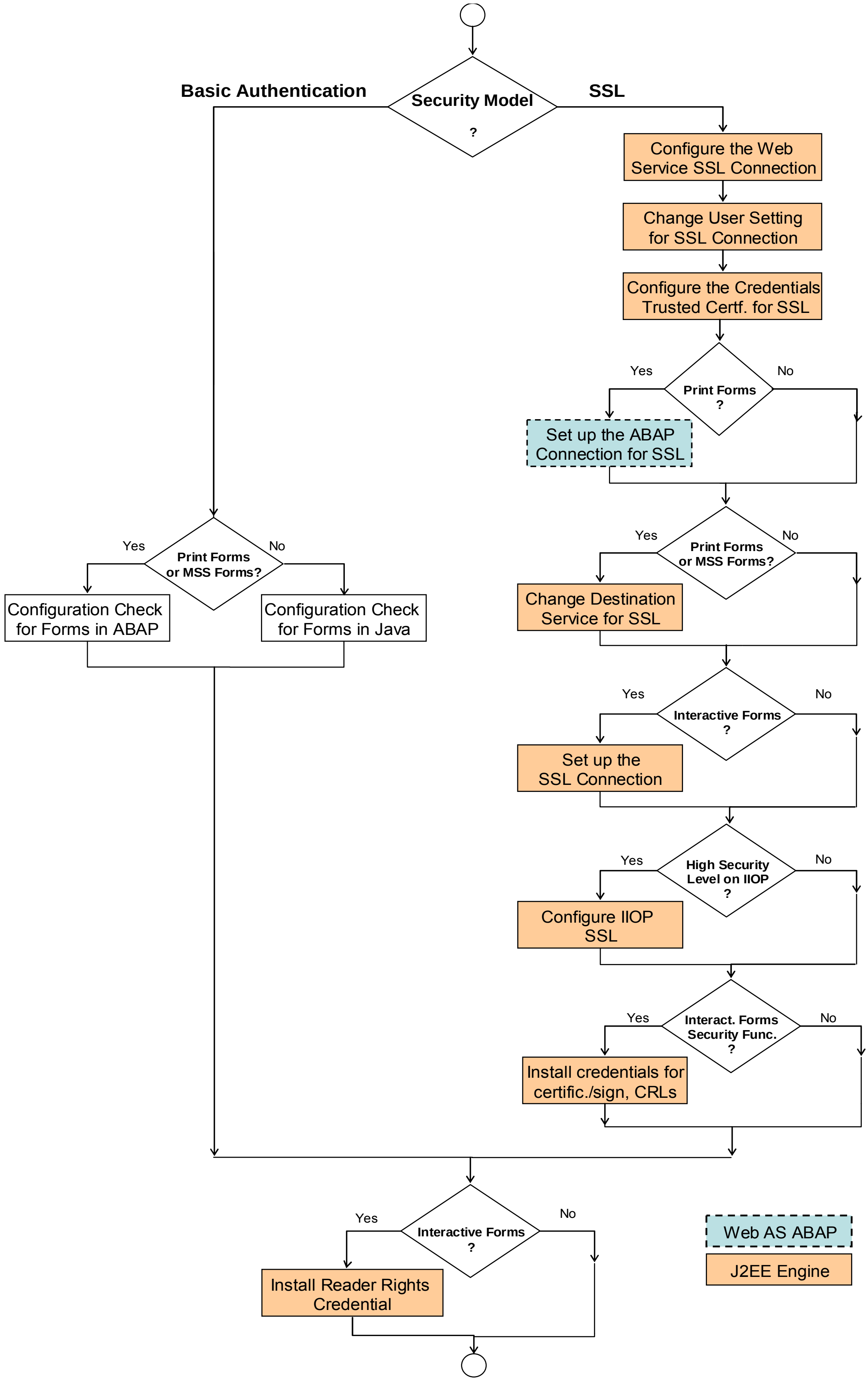
Adobe document services expose their functionality to the PDF object implementations through a Web service interface. This interface is not directly accessible. Instead, access to Adobe document services is provided using either:

- The PDF object, or
- Web Dynpro and Forms Processing, which in turn use the PDF object at runtime

Adobe document services can perform a number of tasks that require access to sensitive corporate resources. For example, to assign usage rights to a document, Adobe document services require access to credentials. It is therefore important to ensure that only authorized users and processes can access the Adobe document services Web service. Configuring security on your web services connection ensures the security of your documents and credentials.

For more information about the secure communication links in the Web AS with Adobe document services, see Technical System Landscape [Extern] in the Security Guide for SAP Interactive Forms by Adobe.

Configuration Procedure


Basic Authentication
Security Model ?
SSL
Configure the Web Service SSL Connection
Change User Setting for SSL Connection
Configure the Credentials Trusted Certf. for SSL
Yes
Print Forms ?
No
Set up the ABAP Connection for SSL
Yes
Print Forms or MSS Forms?
No
Configuration Check for Forms in ABAP
Configuration Check for Forms in Java
Yes
Print Forms or MSS Forms?
No
Change Destination Service for SSL
Yes
Interactive Forms ?
No
Set up the SSL Connection
Yes
High Security Level on IIOP ?
No
Configure IIOP SSL
Yes
Interact. Forms Security Func. ?
No
Install credentials for certific./sign, CRLs
Yes
Interactive Forms ?
No
Install Reader Rights Credential
Web AS ABAP
J2EE Engine

All steps that are necessary for interactive forms also apply to forms in ISR scenarios, for example MSS Forms.

The following checklist gives you a summary of the information shown in the figure above. Some steps depend on your application scenario. Each step provides a link to the appropriate section in this document.

1. Select a security model for your web services connection and configure the web service connection appropriate to that model. There are two security configuration options:

   - Basic Authentication – supports all functions including assigning usage rights, but excluding digital signatures. If you use this model, you normally do not need to configure the connection settings, because as of SAP NetWeaver 7.0, these are the default settings in an ABAP+Java system. You should check the configuration settings to verify that they are correct for your system. For example, if you changed the client of the ABAP system, you need to check the settings for the client information. For more information, see Configuration Check [Page 11].

   - SSL – supports all functions including digital signatures. For detailed information, see Configuration of the Web Service SSL Connection [Page 23]

     - If you are developing or processing forms in ABAP, you must change the Destination Service for SSL [Page 28], because data is transmitted between ABAP and Java environments. You also must change the setting of the ABAP connection to SSL [Page 27].

     - You need to configure the IIOP SSL [Page 30] only if you require a high level of security.

     - Install and configure the credentials required by SAP Web AS to assign the credentials and CRLs required for document certification and signing. For detailed information, see Installing and Configuring Credentials [Page 33] .

2. If you use interactive forms, install and configure the credentials required by SAP Web AS to assign usage rights (Reader Rights). For more information, see Installing and Configuring Credentials [Page 33] and Reader Rights Credential [Page 33].

3. If your system configuration uses the System Landscape Directory (SLD), you must publish the Adobe document services to the SLD. For detailed information, see Publishing Adobe Document Services to the System Landscape Directory [Page 32].

5.1 Securing Access to the Web Service

To ensure secure access to the Web service of Adobe document services, you can configure the Web service to use one of two security access methods:

- Basic Authentication
  In an ABAP+Java system, Basic Authentication is set up automatically as the default during the initial installation of Adobe document services.

  You must check the settings using the Configuration Check [Page 11] to ensure the configuration is appropriate for your system.

- SSL Connection
  Necessary for SAP applications to digitally sign PDF documents (in addition to

rendering documents and assigning usage rights to PDF documents). For detailed instructions, see Configuration of the Web Service SSL Connection [Page 23].

5.2 Configuration Check

Purpose

Adobe document services (ADS) can run in different IT scenarios, infrastructures and usage types in a new or in an upgraded installation. In some cases the installation process cannot perform all configuration settings that are necessary for the use of Adobe document services, for example, if ABAP and Java are not installed on the same server. Use this process to check whether all configuration steps are complete and to verify, which ones you still need to perform manually.

Prerequisites

Adobe document services are installed on your system.

Process Flow

Depending on your application, choose one of these options:

- Check the configuration for PDF-based print forms or forms in Web Dynpro for ABAP [Page 11]
- Check the configuration for interactive forms in Web Dynpro for Java [Page 15]

5.2.1 Configuration Check for PDF-Based Forms in ABAP

Purpose

Use this process to check whether all configuration steps for developing and processing forms in an ABAP environment are completely and to verify, which ones you still need to perform manually.

Process Flow

1. The first check you need to perform is Checking by Executing Test Report FP_TEST_00 [Page 11]

   The following steps are only necessary, if the result of the above test was not successful.

2. Check the ABAP Connection [Page 12]
3. Check the User and Password [Page 16]
4. Check the Destination Service [Page 14]

5.2.1.1Checking by Executing Test Report FP_TEST_00

Use

This test report checks if your system is configured correctly for processing forms in an ABAP environment.

Prerequisite

A device type for printing PDF-based forms is configured. For more information, see SAP Printing Guide (BC-CCM-PRN) [Extern]

Procedure

1. Log on to your SAP NetWeaver AS ABAP.
2. Call transaction SA38 and enter the name **FP_TEST_00**.
3. Choose *Execute (F8)*. A dialog box is displayed.
4. Enter **FP_TEST_00** in the field *Form.*

   This is displayed as the default form name.
5. Enter the name of the connection to the ADS. Enter the default name ADS, or, if you have specified another name, the one you are using in your system.
6. In the dialog box, choose *Output in Print Preview.*
7. Enter an appropriate device type in the field *Output Device*.
8. Choose *Print Preview*.

Result

If the configuration is correct, a form containing several lines on two pages is displayed.

If the configuration is not correct, no form is displayed. In that case, you need to perform further tests.

See also:

Checking the ABAP Connection [Page 12]

Checking the User and Password [Page 16]

Checking the Destination Service [Page 14]

5.2.1.2Checking the ABAP Connection

Use

This is a test for checking the RFC destination. This test applies to both connections using Basic Authentication and SSL connections.

Procedure

1. Log on to your SAP system.
2. Call transaction SA38.
3. Enter the name of the test report **FP_PDF_TEST_OO**.
4. Enter the name of the connection. Enter the default name **ADS**, or, if you have specified another name, the one you are using in your system.
5. Choose *Execute (F8)*.

Result

If the configuration is correct, the system displays the version number of the Adobe document services.

If the configuration is not correct, the system displays a corresponding message. For solving the problem, see

Setting Up Basic Authentication in an ABAP Environment - Creating the ABAP Connection [Page 18] or

Setting Up the SSL Connection in an ABAP Environment [Page 27].

5.2.1.3Checking the User and Password

Use

This is a test for checking whether the entries for the user, security role, and passwords are correct in a system that uses Basic Authentication.

Procedure

1. Enter the following URL in your Web browser: **`http://<server>:<port>/AdobeDocumentServices/Config`** where **`<server>`** is the name of the J2EE engine where the Adobe document services are installed, and **`<port>`** is the port of the J2EE engine.

   Note that the entries in the URL are case-sensitive.

2. The Web page of the Web service *AdobeDocumentServices* is displayed. Choose *Test*.
3. Choose *rpdata(test.…)*.
4. Choose the *Send* button without entering any parameters.
5. Enter the user name and password provided in the previous configuration steps.
6. Choose *Submit*.

Result

If the configuration is correct, the system displays the version number of the Adobe document services in the response area.

You can ignore the message `Required stream: "PDFDocument" not found.`

If the configuration is not correct, the page does not change and *Submit* remains on the screen. To further check the configuration, perform the following procedures:

PDF-based forms and forms in Web Dynpro for ABAP

- Creating a User in the SAP NetWeaver AS ABAP for Basic Authentication [Page 17]
- Setting Up Basic Authentication in an ABAP Environment - Creating the ABAP Connection [Page 18]

Interactive forms in Web Dynpro for Java

- Creating a User in the SAP NetWeaver AS Java for Basic Authentication [Page 21]
- Setting Up Basic Authentication in a Java Environment [Page 23]

5.2.1.4Checking the Destination Service and the ICF Service

Use

Use the first of the following tests for checking the settings of the destination service. If you receive any error messages, continue with the further tests listed below.

Prerequisites

You have already checked the ABAP connection [Page 12].

1. Checking the Destination Service using a test report in AS ABAP

Using this test you can check if the complete configuration of the destination service and the ICF service is correct. There are two test reports available:

- FP_CHECK_DESTINATION_SERVICE
  This report checks the configuration of the destination service and the ICF service *sap/bc/fp*.
- FP_CHECK_HTTP_DATA_TRANSFER
  This report checks the configuration of the destination service and the ICF service *sap/bc/fpads*. You only need this service, if AS ABAP and AS Java (with ADS) are installed in different systems, and you want to bundle your forms to improve performance.

  The ICF service *sap/bc/fpads* and the corresponding settings in the destination service are not configured during the initial installation of SAP NetWeaver 7.0. For more information, see Creating the Service User ADS_AGENT in the ABAP Environment [Page 21] and Activating the ICF service [Page 20].

1. Log on to your SAP NetWeaver AS ABAP system.
2. Call transaction SA38.
3. Enter the name of the test report.
4. Execute the test without choosing the option *With Destination Service.*
5. The system renders a test form in the background without using the destination service and displays the size of the created PDF.
6. Execute the test again. Now select the option *With Destination Service.*
7. If the settings of the destination service are correct, the system displays the same message as before (see step 5).

2. Checking the ICF Service using the Web Browser

1. In your Web browser enter the URL

   `http://<server>:<port>/sap/bc/fp/form/layout/fp_test_00.xdp`

   where <server> is the server that hosts the AS ABAP and <port> is the http port of the AS ABAP.
2. In the dialog box enter ADS_AGENT as user and the password you have specified for it.

3. If the settings of the ICF service are correct, the system displays the layout information of the form FP_TEST_00 in XML format.

3. Checking the Destination Service using the Visual Administrator

Using this test you can check if the settings for the destination service are correct.

1. Start the Visual Administrator and navigate to *Services* → *Destinations* as described in Creating or Changing the Destination Service [Page 28].
2. Choose the destination and extend entry in the field *URL* you specified in *Connection Settings* to
   `http://<server>:<port>/sap/bc/fp/form/layout/fp_test_00.xdp`

   where `<server>` is the server that hosts the AS ABAP and `<port>` is the http port of the AS ABAP

   ⚠

   Specify the URL exactly as given, otherwise the connection cannot be tested and you receive the error message `Error during ping operation: Received HTTP response 401.`
3. Choose *Save and Test*
4. The system sends a call to the ABAP system where the form layout is stored. If the settings are correct, the system displays the message `HTTP GET response code 200 Content type text/xml`.
5. Don't forget to change the URL back to **`http://<server>:<port>`** for the Destination Service to work properly.
6. *Save*

Further Checks

If the settings of the destination service are not correct, you get an error message. You need to perform further checks:

1. Check if the ICF service [Page 20] is active.
2. Check if the system user ADS_AGENT [Page 21] is correctly configured.
3. Check the settings of the Destination Service [Page 28].

5.2.2 Configuration Check for Interactive Forms in Web Dynpro for Java

Purpose

Using this process you can check the configuration for basic authentication on the J2EE Engine, where the Adobe document services are installed. This configuration is necessary to develop and run interactive forms in Web Dynpro for Java. It does not include any check of the credentials that are needed for interactive forms, for example, credentials for Reader Rights or for digital signatures.

Process Flow

To check, proceed a follows:

- Check the User and Password [Page 16]

- Check the settings for basic authentication [Page 23]

5.2.2.1Checking the User and Password

Use

This is a test for checking whether the entries for the user, security role, and passwords are correct in a system that uses Basic Authentication.

Procedure

1. Enter the following URL in your Web browser: **`http://<server>:<port>/AdobeDocumentServices/Config`** where **`<server>`** is the name of the J2EE engine where the Adobe document services are installed, and **`<port>`** is the port of the J2EE engine.

   Note that the entries in the URL are case-sensitive.

2. The Web page of the Web service *AdobeDocumentServices* is displayed. Choose *Test*.
3. Choose *rpdata(test….)*.
4. Choose the *Send* button without entering any parameters.
5. Enter the user name and password provided in the previous configuration steps.
6. Choose *Submit*.

Result

If the configuration is correct, the system displays the version number of the Adobe document services in the response area.

> You can ignore the message `Required stream: "PDFDocument" not found.`

If the configuration is not correct, the page does not change and *Submit* remains on the screen. To further check the configuration, perform the following procedures:

PDF-based forms and forms in Web Dynpro for ABAP

- Creating a User in the SAP NetWeaver AS ABAP for Basic Authentication [Page 17]
- Setting Up Basic Authentication in an ABAP Environment - Creating the ABAP Connection [Page 18]

Interactive forms in Web Dynpro for Java

- Creating a User in the SAP NetWeaver AS Java for Basic Authentication [Page 21]
- Setting Up Basic Authentication in a Java Environment [Page 23]

5.3 Configuration of the Web Service for Basic Authentication

Purpose

As of SAP NetWeaver 7.0, Basic Authentication is set up automatically as the default during the initial installation. However, there are system constellations where the installation procedure cannot perform all steps completely, for example, if ABAP and Java run on different servers.

Prerequisites

You have checked the configuration status after installation. See Configuration Check [Page 11].

Process Flow

Depending on your scenario, you need to configure Basic Authentication in an ABAP or in a Java system:

ABAP

- Creating a User in the SAP NetWeaver AS ABAP for Basic Authentication [Page 17]
- Setting Up Basic Authentication in an ABAP Environment - Creating the ABAP Connection [Page 18]
- Creating or Changing the Destination Service [Page 28]

Java

- Creating a User in the SAP NetWeaver AS Java for Basic Authentication [Page 21]
- Setting Up Basic Authentication in a Java Environment [Page 23]

5.3.1 Creating a User in the SAP NetWeaver AS ABAP for Basic Authentication

Use

When your system is not an ABAP+Java system the user ADSUser was not created during installation. This user is required for the secure communication between the ABAP system and the Java system where the Adobe document services are installed. Proceed the following procedures:

Creating a User in the SAP NetWeaver AS ABAP:

1. Log on to the SAP system with an admin user, in the client that is used for the UME authentication.
2. Choose *Tools* → *Administration* → *User Maintenance* → *User* (transaction SU01).
3. Enter **`ADSUser`** as user name and choose Create.
4. Choose *system user* as type for ADSUser.
5. Enter a password and save your settings.

Assigning a Role in ABAP

1. Choose *Tools* → *Administration* → *User Maintenance* → *Role Administration* → *Roles* (transaction PFCG)
2. Create a role **`ADSCallers`** (no authorizations required).

The ADSCallers role in SAP NetWeaver AS ABAP appears automatically as the ADSCallers group in the J2EE Engine.

3. Activate the role.
4. Assign user *ADSUser* to this role.

Assigning the Security Role in Java

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57])
2. On the Cluster tab, choose *Server <x>* → *Services* → *Security Provider*.
3. On the *User Management* tab, choose the *Tree* tab in the left panel.
4. In the *User* Tree, ensure that the user you created in ABAP appears under the *ADSCallers* group.
5. On the *Policy Configurations* tab, in the *Components* area, select *com.adobe/AdobeDocumentServices*AdobeDocumentServicesAssembly.jar.*
6. On the *Security Roles* tab, select *ADSCaller* from the *Security Roles* list.
7. In the *Mappings* area, choose *Add*, which is assigned to *Users*. A dialog *Choose Users or Groups* is displayed.
8. Choose the *Tree* tab.
9. In the *User Tree*, under the *ADSCallers* group, select the *ADSUser* you just created and choose *OK*. This assigns the new user to the *ADSCaller* security role.

5.3.2 Setting Up Basic Authentication - Creating the ABAP Connection

Use

This procedure applies only in the scenario of print forms or forms created in an SAP Web AS ABAP. The purpose of this procedure is to create a connection in the ABAP environment to use when connecting to Adobe document services and to set up Basic Authentication.

Procedure

1. Log on to your SAP Web AS central instance host.
2. Call transaction SM59.
3. Choose *Create*.
4. Enter at least the following:

   RFC destination: **ADS**

   Connection type: **G**

   Description: <your description>
5. Choose ENTER

6. Choose the *Technical settings* tab and enter at least the following:

   Target Host

   Enter the host name of the J2EE Engine that runs the Adobe document services or of the SAP Web dispatcher if applicable.

   Service No

   Enter the HTTP port number of the Target Host you have specified (The following naming convention applies: **`5<J2EE_instance_number>00`** (5**00**00, for example, if your J2EE instance is **`00`**).

   Path Prefix

   Enter exactly the string **`/AdobeDocumentServices/Config?style=rpc`**

   A warning is displayed: `Query String Not Allowed.` Ignore this warning by pressing **`Enter`**.

7. Choose the *Logon/Security* tab, select *Basic Authentication*.
8. In the *User* and *Password* boxes, enter the user name **`ADSUser`** and the password.
9. Save your settings.
10. Choose *Test Connection*.
11. A screen is displayed. The field *status_reason*: `OK` indicates that the test was successful.

5.3.3 Creating or Changing the Destination Service

Use

This procedure applies to SAP applications using print forms and forms in ISR scenarios, for example in the Business Package Manager Self-Services.

When processing forms between an ABAP environment and a Java environment, the Destination service of the J2EE engine is used. This service runs in the Java environment and facilitates communication and data transmission between the Java and ABAP environments. Communication between ABAP and the Java Destination service is enabled by the Internet Communication Framework (ICF).

In an ABAP+Java system, the Destination service is configured for basic authentication, when the system is installed. You need to change this configuration, if

- ABAP and Java are installed on separate systems
- you want to use SSL
- the service user ADS_AGENT in the ABAP system is not created in the default client.

Prerequisites

- The ICF service on the SAP NetWeaver AS ABAP is active. For more information, see Activating the ICF service [Page 20].

- The service user ADS_AGENT was created and assigned to the appropriate role. For more information, see Creating the Service User ADS_AGENT in the ABAP Environment [Page 21]

Procedure

To change destination, proceed as follows:

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)
2. On the Cluster tab, choose *Server <x>* → *Services* → *Destinations*. Under *Runtime*, select *HTTP*. The available destinations are displayed.
3. Choose the destination you want to change. The information that applies to a selected destination is displayed in the right pane. Proceed with step 6.
4. Choose *New* in the navigation panel, if the destination does not exist, yet.
5. In the dialog box that follows, enter the name **`FP_ICF_DATA_<SID>`**, where <*SID*> is your ABAP system, for the new destination and choose OK.
6. Under *Connection Settings*, enter the message server (or Web Dispatcher) of the SAP NetWeaver AS ABAP in the *URL* field:

   `http://<hostname>:<HTTP_port>`

   and in case of SSL,

   `https://<hostname>:<HTTPS_port>`

   To display the host name of your SAP NetWeaver AS ABAP log on to SAP NetWeaver AS ABAP and call transaction SICF. In the main menu, choose *Goto* → *Port Information*. The information is displayed on a screen; where the HTTP_port is specified under *Services*.

7. Enter the appropriate client number of the system, where the service user ADS_AGENT exists into the *client* field. Keep the other fields *System ID* and *Language* empty.
8. Under *Logon Data*, choose *BASIC* in the *Authentication* field.
9. Enter **`ADS_AGENT`** in the *Username* field and enter the same password as given for this service user in the SAP NetWeaver AS ABAP in the *Password* field.
10. Choose *Save.*

    If you choose *Save and Test*, you can ignore the message *Error during ping operation: Received HTTP response 404.*

5.3.4 Activating the ICF service

Use

The communication between the Destination Service of the SAP Web AS Java and the SAP Web AS ABAP uses the Internet Communication Framework. You have to activate the corresponding service, if this is not done so far.

Procedure

1. Log on to your SAP Web AS ABAP system
2. Choose transaction SICF.
3. Choose *default_host* → *sap* → *bc* → *fp* in the tree.
4. Choose *Service/Virt.Host* → *Activate*
5. If AS ABAP and AS Java (with ADS) are installed in different systems, and you want to bundle your forms to improve performance, you also have to activate the *fpads* ICF service.

Result

The ICF service is now active.

5.3.5 Creating the Service User ADS_AGENT in the ABAP Environment

Use

The service user ADS_AGENT in the ABAP environment corresponds to the user you specify in the authentication parameters of the Destination Service in the AS Java where the Adobe document services are installed.

Procedure

1. Log on to the SAP NetWeaver AS ABAP and choose transaction SU01 (User Management).

   You must specify this client in the Destination Service.

2. Enter the name **ADS_AGENT** in the *User* field and choose *User* → *Create*.
3. Choose the *Logon data* tab and assign a password.

   You must specify this password in the Destination Service.

4. Choose *Service* as the user type for ADS_AGENT.
5. Choose the *Role* tab and assign one of the following roles to the user ADS_AGENT.

   ○ *SAP_BC_FP_ICF*,
   if AS ABAP and AS Java (with ADS) are on the same system (Double-Stack or Java Add-In).

   ○ *SAP_BC_FPADS_ICF*,
   if AS ABAP and AS Java (with ADS) are on different systems.

   This distinction is for performance reasons. You may copy the required role first. For more information, see Changing Standard Roles [Extern] in the SAP Library.

6. Save the data.
7. Create or change the Destination Service [Page 28]

See also:

Bundling of PDF-Based Forms [Extern]

5.3.6 Creating a User in the SAP NetWeaver AS Java for Basic

Authentication

Use

In some cases the user ADSUser was not created during installation. This user is required for the secure communication between the Web Dynpro application and the Java system where the Adobe document services are installed.

You can create this user in the J2EE Engine or in the SAP NetWeaver AS ABAP depending on the J2EE installation settings for the SAP User Management Engine (UME). You create this user in the SAP NetWeaver AS ABAP [Page 17] when the UME is configured against the ABAP backend.

Creating the User ADSUser and Assigning the Security Role

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57])
2. On the *Cluster* tab, choose *Server <x>* → *Services* → *Security Provider.*
3. On the *User Management* tab, choose *Create Group* to create a group called **`ADSCallers`**, if the group does not exist. In the dialog that follows, enter the name and choose *OK*.
4. Choose *Create User*. The *Create New User* dialog box is displayed.
5. In the *User name*, *Password*, and *Confirm password* boxes, enter **`ADSUser`** for the user name and type a password.
6. Choose the *Tree* tab in the right panel. In the *User Tree*, select *ADSCallers,* and then choose *OK*.
7. On the *Policy Configurations* tab, in the *Components* area, select *com.adobe/AdobeDocumentServices*AdobeDocumentServicesAssembly.jar*.
8. On the *Security Roles* tab, select *ADSCaller* from the *Security Roles* list.

   The ADSCaller security role was created when your system was installed. You should not assign this security role to users other than the system user that you will use for accessing Adobe document services.

9. In the *Mappings* area, choose *Add,* which is assigned to *Users*. A dialog *Choose Users or Groups* is displayed.
10. Choose the *Tree* tab.
11. In the *User Tree*, under the *ADSCallers* group, select the *ADSUser* you just created and choose *OK*. This assigns the new user to the ADSCaller security role.

Defining ADSUser as Technical User

To prevent, that the password for the ADSUser expires, do following:

1. Log on to the *User Management* of your AS Java:

   `http://<AS Java server>:<http port>` → *User Management*

2. Display the user ADSUser.
3. Choose *Modify*.
4. In the *Security policy* field, change the setting from *Default* to *Technical User*
5. Save.

5.3.7 Setting Up Basic Authentication in a Java Environment

Use

This procedure applies for the scenario of interactive forms. Set up Basic Authentication to access the Java version of the PDF object. This procedure describes you the configuration steps and applies when the Adobe document services and the Web Dynpro runtime are installed on the same J2EE Engine. Then under *Destination* the *URL* is set to *Default*.

Prerequisites

The user ADSUser was created and configured during installation of Adobe document service.

Procedure

To set up Basic Authentication in a Java environment:

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57])
2. On the Cluster tab, choose *Server <x>* → *Services* → *Web Services Security*.
3. Choose *Web Service Clients* → *sap.com > tc~wd~pdfobject* → *com.sap.tc.webdynpro.adsproxy.AdsProxy*ConfigPort_Document*.
4. From the *Authentication* list, select *BASIC*.
5. In the *User* and *Password* boxes, enter as Username **`ADSUser`** and a Password.
6. Choose *Save*.
7. The authentication data must be activated. For doing this navigate to *Services* → *Deploy*.
8. Choose the button *Application*.
9. Choose *sap.com/tc~wd~pdfobject* in the tree.
10. Choose *Stop Application*.
11. For restarting the application choose *Start Application*.

> If the Adobe document services and the Web Dynpro runtime environment are not installed on the same J2EE Engine, you have to configure a *Custom* URL. For more information, see Configuring the Destination URL for the Adobe Document Services [Extern] in the SAP Library.

5.4 Configuration of the Web Service SSL Connection

Purpose

To perform security-related functions such as digitally signing PDF documents, you must set up an SSL connection to the Web service.

> If you use these security-related functions in forms in Web Dynpro for Java, you also have to configure the SSL connection for Web Dynpro. For more information, see SAP Note 838111.

Process Flow

The checklist below summarizes the steps required to configure the Web Service SSL connection:

1. Create a view called ADSCerts in the Key Storage service, which is necessary for the storage of the client certificates for the Adobe document services. For detailed instructions, see Creating a View in the Key Storage Service [Page 25].
2. Set up SSL (a.) and configure the client certificates (b.) for the J2EE Engine where the Adobe document services and where Web Dynpro for Java are installed. You will find the information for these procedures under:
    a. Configuring the Use of SSL on the SAP J2EE Engine [Extern] in the SAP Library under *SAP NetWeaver → Security → Network and Transport Layer Security → Transport Layer Security on the SAP J2EE Engine*

    b. Configuring the Use of Client Certificates for Authentication [Extern] in the SAP Library under SAP *Netweaver → Application Platform (SAP Web Application Server) → Java Technology in SAP Web Application Server → Administration Manual → Server Administration → J2EE Engine Security → Authentication on the J2EE Engine → Configuring Authentication Mechanisms → Using Client Certificates for User Authentication*

    Follow all steps as described in this documentation, except for step 3, because you do not need to configure UME properties and LDAP in this scenario.

    c. Store the client certificates in the ADSCerts view you created earlier.
3. This step is only necessary for the print forms scenario.
    a. Set up SSL and configure the client certificates on the SAP Web AS ABAP. You will find the information for this procedure under: Configuring the SAP Web AS for Supporting SSL [Extern] in the SAP Library under *SAP NetWeaver → Security → Network and Transport Layer Security → Using the Secure Sockets Layer Protocol with the SAP Web AS ABAP*

    b. Import the client certificates into the J2EE Engine where the Adobe document services are installed, as described in step 2b.

    c. Store the client certificates in the ADSCerts view you created earlier.
4. This step is only necessary for the interactive forms scenario in Web Dynpro for Java. See Configuring Web Dynpro User Access to Key Storage [Page 25].
5. Change the settings of the system user ADSUser from basic authentication to SSL for the secure communication. For detailed instructions, see Configuring the User ADSUser for the SSL Connection [Page 26].
6. In an ABAP environment, set up an SSL connection between the ABAP connection and the J2EE environment. For detailed instructions, see Setting Up the SSL Connection in an ABAP Environment [Page 27].
7. In a Web Dynpro for Java environment, configure an SSL connection between the Java version of the PDF object and the Adobe document services. For detailed instructions, see Setting Up the SSL Connection in a Java Environment [Page 29].
8. This step is only necessary for scenarios that require high security within SAP Web AS. Adobe document services are installed on the J2EE Engine and consist of two parts. The communication between these parts uses the IIOP service. If you need to set up SSL on this communication path, proceed as follows:
    a. Download and deploy the BinariesSSL-2 Library. This library contains strong encryption components and is required for the secure IIOP communication. You may need authorization to receive this library. For more information, see SAP Note 752153.

Note that the functions digital signatures and certification of forms can also be performed even if this library is not used.

b. Configure the IIOP SSL [Page 30].

5.4.1 Creating a View in the Key Storage Service

Use

Client certificates should be imported into a Key Storage view called *ADSCerts*.

Procedure

To create an *ADSCerts* view in the Key Storage service:

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57]).
2. On the *Cluster* tab, choose *Server <x>* → *Services* → *Key Storage*.
3. On the *Runtime* tab, choose *Create View*.
4. In the *Input* dialog box, enter the alias **`ADSCerts`**, and choose *OK*.
5. Configure the user ADSUser for the SSL connection [Page 26].

5.4.2 Configuring Web Dynpro User Access to Key Storage

Use

If the communication uses SSL, all users working in Web Dynpro for Java with SAP Interactive Forms by Adobe need access to the Key Storage. Proceed with the following steps if you want to specify one or some users (group), who are allowed to work with SAP Interactive Forms. Users are all persons who work with SAP Interactive Forms, persons who fill in form fields in a form displayed in the Web Dynpro client, and developers creating a form.

Prerequisites

- The users working with SAP Interactive Forms in Web Dynpro have been created before.
- The view ADSCerts has been created before. For more information, see Creating a View in the Key Storage Service [Page 25].

Procedure

1. Decide which users or groups should have access to the Key Storage.
2. Define a role for all users in step 1. No actions are required. For more information, see Managing Users, Groups, and Roles [Extern].
3. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)
4. On the Cluster tab, choose *Server <x>* → *Services* → *Security Provider.*
5. Under *Runtime* → *Policy Configuration*, choose *keystore-view.ADScerts.*
6. Under *Security Roles* tab, choose *view-creator.*

7. *Add* the user respectively the group from step 1.
8. Restart the cluster, including all J2EE Engines and dispatchers.

5.4.3 Configuring the User ADSUser for the SSL Connection

Use

To ensure secure access, you must change the settings of the system user ADSUser, which is used for accessing the Adobe document services.

Prerequisites

- The security role ADSCaller was created in the J2EE Engine during the initial installation.

  ⚠

  You should not assign this security role to users other than the system user that you will use for accessing Adobe document services.

  💡

  You can see this role in the Visual Administrator under *Server <x>* → *Services* → *Security Provider* → *Runtime* → *Policy Configurations*. Choose *com.adobe/AdobeDocumentServices*AdobeDocumentServicesAssembly.jar* in the *Components* area to display the role in the *Security Roles* tab.

- ADSUser was created during the initial installation of SAP NetWeaver and is assigned to the security role ADSCaller.

  💡

  ADSUser already exists in the J2EE Engine [Page 21] or in the SAP NetWeaver AS ABAP [Page 17] depending on the J2EE installation settings for the SAP User Management Engine (UME). It is in the SAP NetWeaver AS ABAP when the UME is configured against the ABAP backend. In this case, also a role ADSCallers was created in the SAP NetWeaver AS ABAP and ADSUser was assigned to this role. The ADSCallers role in SAP NetWeaver AS ABAP appears automatically as the ADSCallers group in the J2EE Engine.

- The group ADSCallers in the J2EE Engine exists.

Procedure

You change the user settings in the J2EE Engine.

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)
2. On the *Cluster* tab, choose *Server <x>* → *Services* → *Security Provider*.
3. Choose *User Management* and then the *Tree* tab in the left panel.
4. Select *ADSCallers* → *ADSUser*.
5. In the *Authentication* area, choose *Add*.
6. In the *Add Certificates* dialog box, from the *Select view* drop-down list box, select the *ADSCerts* view.
7. From the *Select entries* list, select the certificate that you want to associate with this user, and then choose *OK*.

8. In the *Authentication* area, select *No password change required*.

Result

The user ADSUser is configured for the use of SSL.

Dependent on your scenario you have to set up the SSL connection in the ABAP or in the Java Environment.

See also:

Setting Up the SSL Connection in an ABAP Environment [Page 27]

Setting Up the SSL Connection in a Java Environment [Page 29]

5.4.4 Configuring the Credentials and Trusted Certificates to Use SSL

Procedure

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)
2. On the *Cluster* tab, choose *Server <x>* → *Services* → *SSL Provider*.
3. On the *Runtime* tab, select *Dispatcher <x>*.
4. Under *Configuration*, select *Active Sockets*, then select the host with the port set to 5xx01.
5. On the *Client Authentication* tab, select *Request client certificate*, then choose *Add*.
6. Select the certificate from the *Available Credentials* dialog box, then choose *OK*.

   This certificate is also located under the *TrustedCA* view in Key Storage.

5.4.5 Setting Up the SSL Connection in an ABAP Environment

Use

In an ABAP environment, you need to set up the SSL connection between the ABAP and the J2EE environments.

If the ABAP Connection to the Adobe document services does not yet exist, see Setting Up Basic Authentication - Creating the ABAP Connection [Page 18].

Procedure

To set up the SSL connection in an ABAP environment:

1. Log on to your SAP system and go to transaction SM59.
2. In the *RFC Destinations* tree, select *HTTP Connections to Ext. Server*.
3. Select *ADS*, then choose *Change*.
4. On the *Logon/Security* tab, in the *SSL* area, select *SSL Client Certificate*.
5. Select the certificate.
6. Select *Active*.

7. On the *Technical Settings* tab, in the *PathPrefix* box, enter **`/AdobeDocumentServicesSec/Config?style=rpc`**

8. Choose *Save*.

5.4.6 Creating or Changing the Destination Service

Use

This procedure applies to SAP applications using print forms and forms in ISR scenarios, for example in the Business Package Manager Self-Services.

> When processing forms between an ABAP environment and a Java environment, the Destination service of the J2EE engine is used. This service runs in the Java environment and facilitates communication and data transmission between the Java and ABAP environments. Communication between ABAP and the Java Destination service is enabled by the Internet Communication Framework (ICF).

In an ABAP+Java system, the Destination service is configured for basic authentication, when the system is installed. You need to change this configuration, if

- ABAP and Java are installed on separate systems
- you want to use SSL
- the service user ADS_AGENT in the ABAP system is not created in the default client.

Prerequisites

- The ICF service on the SAP NetWeaver AS ABAP is active. For more information, see Activating the ICF service [Page 20].
- The service user ADS_AGENT was created and assigned to the appropriate role. For more information, see Creating the Service User ADS_AGENT in the ABAP Environment [Page 21]

Procedure

To change destination, proceed as follows:

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)
2. On the Cluster tab, choose *Server <x>* → *Services* → *Destinations*. Under *Runtime*, select *HTTP*. The available destinations are displayed.
3. Choose the destination you want to change. The information that applies to a selected destination is displayed in the right pane. Proceed with step 6.
4. Choose *New* in the navigation panel, if the destination does not exist, yet.
5. In the dialog box that follows, enter the name **`FP_ICF_DATA_<SID>`**, where <*SID*> is your ABAP system, for the new destination and choose OK.
6. Under *Connection Settings*, enter the message server (or Web Dispatcher) of the SAP NetWeaver AS ABAP in the *URL* field:

   `http://<hostname>:<HTTP_port>`

   and in case of SSL,

`https://<hostname>:<HTTPS_port>`

> 💡
>
> To display the host name of your SAP NetWeaver AS ABAP log on to SAP NetWeaver AS ABAP and call transaction SICF. In the main menu, choose *Goto* → *Port Information*. The information is displayed on a screen; where the HTTP_port is specified under *Services*.

7. Enter the appropriate client number of the system, where the service user ADS_AGENT exists into the *client* field. Keep the other fields *System ID* and *Language* empty.
8. Under *Logon Data*, choose *BASIC* in the *Authentication* field.
9. Enter **`ADS_AGENT`** in the *Username* field and enter the same password as given for this service user in the SAP NetWeaver AS ABAP in the *Password* field.
10. Choose *Save.*

> 💡
>
> If you choose *Save and Test*, you can ignore the message *Error during ping operation: Received HTTP response 404.*

5.4.7 Setting Up the SSL Connection in a Java Environment

Use

You need to configure an SSL connection between the Java version of the PDF object and the Adobe document services Web service.

Procedure

To set up the SSL connection in a Java environment:

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)
2. On the *Cluster* tab, choose *Server <x>* → *Services* → *Web Services Security*.
3. Choose *Web Services Clients* → *sap.com > tc~wd~pdfobject* □ *com.sap.tc.webdynpro.adsproxy.AdsProxySec*ConfigPort_Document*.
4. Change the URL to **`https://<Host>:<Port>/AdobeDocumentServicesSec/Config?style=document`**.
5. From the *Authentication* drop-down list box, select *X.509 Client Certificate*.
6. In the *Client Certification Authentication* area, from the *Keystore* view list, select *ADSCerts*.
7. From the *Certificate* list, select the certificate associated with the user that is assigned the ADSCaller security role, which you created earlier.
8. Choose *Save*.

5.4.8 Configuration of the IIOP SSL

Purpose

The Adobe document services consist of two parts, both installed on the SAP Web AS. The communication between these parts uses the IIOP service. You only need to configure SSL on this communication path, if your scenarios require a high level of security.

Prerequisites

You have configured the Web Service SSL connection [Page 23] as described above. You have downloaded and deployed the BinariesSSL-2 Library.

Process Flow

1. Creating the SSL User Credentials [Page 30]
2. Configuring the Adobe Document Services User Credentials [Page 31]
3. Enabling SSL for Adobe Document Services [Page 31]

5.4.8.1Creating the SSL User Credentials

Procedure

To create the SSL user credentials:

1. Log on to the Visual Administrator (See How to Start the Visual Administrator [Page 57]).
2. On the *Cluster* tab, choose *Server <x>* → *Services* → *Key Storage*.
3. On the *Runtime* tab, under *Views*, select *service_ssl*.
4. In the *Entry* area, choose *Create*.
5. In the *Key and Certificate Generation* dialog box, enter the *Subject Properties*, for example:

   Sample Subject Properties

Subject Property	Sample User Entry
Country Name:	`US`
State/Province:	`Some State`
Locality Name:	`Some City`
Organization Name:	`Some Customer`
Organization Unit Name:	`Some Purchasing Unit`
Common Name:	`localhost`

6. In the *Entry Name* box, enter **`ads-credentials`** (exactly as shown).
7. Select *Store Certificate*, then choose *Generate*.
8. Under *Views*, select *TrustedCAs*.
9. In the *View* area, choose *Import from Other*.
10. In the *Select entries to import* dialog box, select *service_ssl* from the *Select view* drop down list box.
11. Under *Select entries*, select *ads-credentials-cert*, and then choose *OK*.

12. Configure the Adobe document services user credentials [Page 31].

5.4.8.2Configuring the Adobe Document Services User Credentials

Procedure

To configure the user credentials:

1. In the Visual Administrator, on the *Cluster* tab, choose *Server <x>* → *Services* → *SSL Provider*.
2. On the *Runtime* tab, select *Dispatcher <x>*.
3. Under *Configuration*, select *Active Sockets*, then select the host with the port set to 50003, (or 50103 if your server instance is 1, 50203 if server instance is 2, and so on).
4. On the *Server Identity* tab, choose *Add*.
5. In the *Available Credentials* dialog box, select *ads-credentials*, then choose *OK*.
6. On the *Client Authentication* tab, select *Require client certificate*, then choose *Add*.
7. Select *ads-credentials-cert*, then choose *OK*.
8. Under *Configuration*, select *New Sockets*.
9. On the *Server Identity* tab, choose *Add*.
10. In the *Available Credentials* dialog box, select *ads-credentials*, then choose *OK*.
11. On the *Client Authentication* tab, select *Require client certificate*, then choose *Add*.
12. In the *Available Credentials* dialog box, select *ads-credentials-cert*, then choose *OK*.
13. Enable SSL [Page 31] on SAP Web AS.

5.4.8.3Enabling SSL for Adobe Document Services

Procedure

To enable SSL:

1. In the Visual Administrator, on the *Cluster* tab, choose *Server <x>* → *Services* → *Document Services Data Manager*.
2. On the *Properties* tab, select *EnableSSL*.
3. In the *Value* box, change the property from **`false`** (the default) to **`true`**.
4. Choose *Update*.
5. Save the changes.
6. When prompted to restart Service Document Services Data Manager, choose *Yes*.
7. Restart SAP Web AS for the change to take effect.

6 Publishing the ADS to the System Landscape Directory

Use

If you have installed the Adobe document services and the Web Dynpro runtime environment on different J2EE Engines and if the communication between these engines uses the System Landscape Directory (SLD), you need to publish the Adobe document services to the SLD.

Prerequisites

A System Landscape Directory (SLD) must have already been configured.

If you have not yet done so, perform the necessary activities according to the documentation *SAP System Landscape Directory*, *Administrative Activities* section, chapters:

- *Start and stop the SLD service*
- *Configure the SLD server*
- *Configure data persistence*
- *Make settings for the SLD bridge*

You can find this documentation by calling the following URL in your web browser (you must have an SLD administrator account):

`http://<host>:<HTTP_port>/sld`

where **`<host>`** is the host name of the SLD host and **`<HTTP_port>`** is the HTTP port of the SAP J2EE engine (The following naming convention applies: `5<J2EE_instance_number>00`. 5**00**00, for example, if your J2EE instance is **`00`**).

From the menu, choose *Help*.

Procedure

1. Start the Visual Administrator. (See How to Start the Visual Administrator [Page 57]).
2. On the *Cluster* tab, choose *Server <x>* → *Services* → *Web Services Container*.
3. In the right frame, select the Web service *AdobeDocumentServices* (*Runtime* tab, *Web Services* frame).
4. Choose the *SLD* tab.
5. Choose *Edit*.

   The system automatically fills the fields with the required information.
6. Enter a description and choose *Publish* to publish the Web service *AdobeDocumentServices* to the SLD.

Result

The Web service is now published to SLD.

7 Installing and Configuring Credentials

Adobe document services require access to a credential (also called a private key) in SAP Web AS to assign usage rights to PDF documents. This is typically the Adobe Reader Rights credential.

If you require additional document security such as certification or digital signatures, you can obtain other credentials from a Certificate Authority (CA). You install and configure other credentials the same way that you install the Adobe Reader Rights certificate.

> 💡
>
> Only DER-encoded X.509 certificates are supported.

Each credential is stored in a Public Key Cryptography Standards (PKCS) #12 file, a hardware device known as a Hardware Security Module (HSM), or as an MSCAPI record in the certificate database on your Microsoft Windows system. For Adobe document services, you must install and handle each credential in a special way:

- A PKCS #12 credential may be delivered simply as a PKCS #12 file, with a .pfx filename extension, on a disk, or over the Web. This file is password-protected and must be handled with care because it represents an extremely valuable resource – the identity of the owner. In the Visual Administrator, PKCS #12 credentials are also called P12 Records.
- An MSCAPI credential is stored in the certificate storage database on your Microsoft Windows system. The Certificate Authority that provides credentials can recommend which credentials should be stored in the MSCAPI certificate storage database.

  > ⚠
  >
  > Do not make a duplicate copy of these credential files except for backup purposes. These backups must be stored securely. Normal system backups must never be allowed to back up a credential file.

- An HSM credential is delivered as a hardware device that must be connected to the system. This credential is much more secure than a PKCS #12 credential because once inserted into the device, it cannot be copied from the device. For installations where security is a priority, it is advantageous to copy any PKCS #12 credentials into a HSM where they are more secure. Access to the HSM is password-protected.

In any of these cases, you must install and configure the credentials in Adobe document services. For ease of use throughout the SAP system, the credential is identified by an alias. The alias is simply a text name that represents the credential.

> 💡
>
> On UNIX systems make sure that you enter file names correctly as given in this document, because the corresponding check is case-sensitive.

Adobe document services log messages that warn when a credential is about to expire. You can set the number of days that the server begins logging daily warning messages before the credential expires. Adobe document services checks the credentials daily to calculate which credentials it should log messages for. You can configure the time of day that the expiry dates are calculated. See Configuring Credential Expiry Logging [Page 38].

7.1 Reader Rights Credential

If you want to create interactive forms, you need a Reader Rights Credential (usage rights credential). Adobe provides a free reader (called Adobe Reader) that allows anyone to view PDF documents on virtually any desktop computer. Adobe Reader runs either as a standalone application or inside a web browser.

While Adobe Reader allows users to view PDF documents, many advanced capabilities such as applying digital signatures and saving documents are not allowed. PDF documents can, however, include usage rights that enable users to fill in forms, add comments, and sign documents using Adobe Reader. These usage rights, also called Reader Rights, allow Adobe Reader to perform tasks that normally require Acrobat Standard or Acrobat Professional.

To apply usage rights to an interactive form, the document must be signed with a special credential. The credential is therefore unique to every company.

Because the Reader Rights Credential applies usage rights to documents, but does not certify or sign them, it does not require a corresponding public key that recipients use to validate signed documents.

To obtain your Reader Rights Credential, see SAP Note 736902.

The credential you will receive is a PKCS #12 (.pfx) file that you need to install. For more information, see Installing a PKCS #12 Credential [Page 35].

Use the alias **ReaderRights** for this credential.

7.2 Credentials for Document Certification and Digital Signatures

You obtain your credentials and the corresponding public keys (as part of a certificate) for certifying and signing from a Certification Authority (CA). You need to install the credential as described below.

To apply a server-side certification or a server-side digital signature to a PDF document, the document must be signed with a credential. The credential is therefore unique. The following table gives you an overview of all credentials.

Types of Credentials

Credential is Used for:	Default Alias of the Credential
Digital Signatures	ServerSignature
Certification	DocumentCertification
Reader Rights	ReaderRights

You should use the alias for the credential. If you need to install more than one credential on your system, use the default alias for the default credential and any other alias for additional credentials. SAP applications certifying a PDF document must specify the name of the credential; otherwise, the default credential is used.

For more information about digital signatures and certification of Interactive Forms see the SAP Library under: *SAP Netweaver Library* → *SAP NetWeaver by Key Capability* → *Application Platform by Key Capability* → *Business Services* → *SAP Interactive Forms by Adobe*.

Public Keys and Certificate Revocation Lists (CRLs)

If you are using credentials in addition to the Reader Rights credential from Adobe, you will also need to install and configure Trusted Anchors and CRLs. For more information, see Trusted Anchors and Certificate Revocation Lists [Page 39].

7.3 Installing a PKCS #12 Credential

Use

Once you receive your PKCS #12 credential file, you must install the file in the appropriate location on your file system.

> On UNIX systems, the directories and files that contain the trust configuration information must be accessible by the SAP Web AS admin account, by default **`<SAPSID>adm`**.

Procedure

To install a PKCS #12 file:

1. Copy the credential file (`<filename>.pfx`) to the *`/usr/sap/<SAPSID>/SYS/global/AdobeDocumentServices/ TrustManagerService/trust/credentials`* directory.

   > This directory was created when the Adobe document services were installed. In earlier versions than SAP NetWeaver 04 SPS 12, the procedure for creating the corresponding directory was different. For more information, see SAP Note 682619. If you have imported an Adobe document services patch, see also SAP Note 727168.

2. Repeat this step on each *Server* node. Note that this step is not required on the *Dispatcher* node.

   > If the *Server* nodes are running within a single cluster, the nodes are updated automatically and you do not have to repeat the step.

3. Configure the credential attributes for each credential, such as registering the password, as described in Configuring Credential Attributes [Page 37].
4. Restart the service *PDF Manipulation Module* for the changes to take effect. (See How to Restart a Service [Page 58].)

7.4 Installing an HSM Credential

Use

HSM credentials are stored in an HSM device. Refer to your HSM device documentation for information about installing the HSM credential.

Procedure

1. After you have installed the HSM Credential, you must configure it by specifying the slot where the HSM is connected and the DLL path by which the credential can be accessed. To configure an HSM credential, see Configuring Credential Attributes [Page 37].
2. Restart the service *PDF Manipulation Module* for the changes to take effect. (See How to Restart a Service [Page 58].)

7.5 Installing an MSCAPI Credential

Use

MSCAPI certificates are stored in the Microsoft Windows certificate database. This storage area is accessible through the Internet Explorer *Tools* → *Internet Options* → *Content* menu. When you receive a credential from a CA that you want to keep in the Microsoft Windows certificate database, install the certificate using the Windows Certificate Import Wizard. When you open a certificate file, click the *Install Certificate* button and follow the instructions in the Wizard.

Procedure

1. After you have installed the MSCAPI certificate, you must configure it by specifying a password and alias, and also a sha1 value if required. To configure an MSCAPI credential, see Configuring Credential Attributes [Page 37].
2. Restart the service *PDF Manipulation Module* for the changes to take effect. (See How to Restart a Service [Page 58].)

7.6 Credential Attributes

Prerequisites

Credentials for document certification and digital signatures:

You have configured the Web Service SSL Connection as described in Configuration of the Web Service SSL Connection [Page 23].

Password and alias

To use a credential, you need a password and an alias for the credential. For security reasons, the password must be stored in a location separate from the credential itself. For SAP Web AS, the passwords must be stored in the Secure Storage Service, in an area reserved for the Adobe document services.

Additional credential attributes

In addition to the required password and alias, you can also configure the following optional attributes, depending on the type of credential you have installed.

After installing a credential and registering its password, you must configure the credential so that it can be correctly and securely used by the system. Each credential record specifies the credential type and alias, and the location or filename of the credential. You must specify the information that pertains to each credential that you have installed.

A credential can be one of three types:

- P12 Record
- HSM Record
- MSCAPI Record

Each of these file types has a number of attributes that must also be set. The file types and their attributes are described in the following table:

Credential Attributes

Attribute	Description	P12	HSM	MSCAPI

		Record	Record	Record
Alias	The name by which the credential is known to the PDF Manipulation Module API. The alias value must be unique.	X	X	X
P12	The filename of the credential file (.pfx file).	X	-	-
Sha1	Credential files can contain multiple keys used for various purposes. The file contains a thumbprint or sha1 value that is used to distinguish among different keys. The sha1 value can be obtained from within the credential file. If the thumbprint is not provided, and multiple appropriate keys are available, a CredentialLoginFailure exception is raised.	X	X	X
DLL Path	The path to the library file that implements the PKCS#11 interface for that particular HSM. The DLL Path can point anywhere in the file system. (Although the attribute is called DLL Path, its value can be any type of library file, including library files used for UNIX.)	-	X	-
Slot	The slot number that identifies where the private key is stored in the HSM.	-	X	-

7.6.1 Configuring Credential Attributes

Use

Configuring the credential attributes consists of registering the password and the alias of each credential which is used by Adobe document services, as well as setting other attributes such as the sha1 value.

To register a password and alias for a credential, proceed as described below in the steps 1 to 6. If you want to configure additional credential settings continue with step 9.

Prerequisites

On AIX platforms, you need to have installed a full version of JCE on the J2EE engine that hosts the Adobe document services in the folder (<JRE_HOME>/lib/security). The JCE files are required for extracting data from the credential file. Restart the J2EE engine after the installation of the JCE.

Ask your JDK vender for more details on downloading and installation of the JCE files.

Procedure

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)
2. On the *Cluster* tab, choose *Server <x>* → *Services* → *Document Services Configuration*.

3. On the *Runtime* tab, select *Credentials*.
4. From the *Type* field, select the type of credential you are configuring (P12 Record, HSM Record or MSCAPI Record).

   The fields that become active and available for editing depend on the credential type that you choose.
5. In the *Alias* field, enter the alias of the credential you installed. Enter the following:
   - **`ReaderRights`** when you configure a Reader right credential for usage rights.
   - **`DocumentCertification`** when you configure a credential for certification.
   - **`ServerSignature`** when you configure a credential for digital signatures.

   Entries for the name of the credential are case-sensitive.
6. For a P12 Record, choose *Browse* to search for the name of the credential and then *Select*.
7. In the *sha1* field, enter the sha1 value. This value can be copied from the credential file itself, and is typically a string of numeric and alphabetic characters. (This step is optional if your credential only contains one sha1 value.)

   ⚠

   If you entered **`ReaderRights`** in the *Alias* field, you must not make any entry in the *sha1* field.
8. For an HSM Record, type the Slot and DLL Path value in the corresponding fields.
9. In the *Password* field, enter the password you received together with the credential you installed.
10. *Confirm* the password, and then click *Add*. The page refreshes and the list of registered credentials at the top of the page includes the credential you just added.
11. Restart the J2EE engine.

7.6.2 Configuring Credential Expiry Logging

Use

Adobe document services log notification messages that warn you when a credential is about to expire. You can specify on the Credentials tab the number of days before the expiry date that you want the warning notification to begin to be logged. The server performs credential expiry calculations daily, to account for any updates to credential information. You can specify the time of day that you want the server to calculate the credential expiry dates.

The expiry date of each credential is displayed in the *Expiry* column of the list of credentials on the *Credentials* tab.

You can view the log messages in the server.log file. See Viewing the Logs [Page 54].

Procedure

To set credential expiry information:

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)

2. On the *Cluster* tab, choose *Server <x>* → *Services* → *Document Services Configuration*.
3. On the *Runtime* tab, select *Credentials*.
4. In the *Credential Expiry Check* area of the dialog, enter a number in the *Begin Logging Warnings* box. (This number represents the number of days before the credential expires that warning messages will begin being logged on a daily basis.)
5. In the *Calculate Credential Expiry Status At* box, enter the time (using 24 hour notation) at which the server scans the credentials to calculate their expiry dates.
6. Click *Set* to save the settings.
7. To perform an immediate calculation of credential expiry dates, click *Now*.

Example

If you set the number of days value to 6 and set the calculation time to 16:30, each day the server checks the expiry date of the credential at 16:30 h. When the server calculates the expiry dates six days before the credential is scheduled to expire, it logs a message to the log file, noting that the credential will expire in six days. The next day, when the server calculates the expiry dates, it logs a message noting that the credential will expire in five days. It continues logging daily messages until the credential expires.

7.7 Trusted Anchors and Certificate Revocation Lists

To be able to use the document certification and digital signatures features, you need to install

- Trusted Anchors to enable the server to verify the certification or signature of a document
- Certificate Revocation Lists (CRLs) for identifying credentials that can no longer be trusted

Trusted Anchors must be installed and configured in Adobe document services. Trusted Anchors may be CA certificates or even individual user’s certificates. There are two cases:

- Typically, when you receive a credential from a CA, it contains two keys: a private key and a public key. While you must keep the private key private and saved within your system, you must make the public key available to anyone to whom you will send documents certified or signed using the corresponding private key. In addition to the credential file containing these keys, the CA also provides a certificate containing the CA’s public key. This is known as a Trusted Anchor.
- You need to install a Trusted Anchor in order to trust signatures or certifications that other people applied to documents using their own credentials.

CRLs prevent you from applying a digital signature that is no longer valid, and it lets you know when digital signatures on incoming documents are invalid. CRLs should be updated on a regular basis (for example, daily or weekly).

7.7.1 Installing Trusted Anchors

Use

To enable the server to verify the certification or signature of a document, you need to install and configure the corresponding Trusted Anchor. This procedure is necessary for documents that are certified or signed by the server as well as documents submitted by users. Trusted Anchors must exist for all CA certificates used to issue credentials including those of the server.

When you install the Trusted Anchor, typically a `.cer` file, you must specify the security-related activities that certificates are trusted for. By doing this, you specify the behavior that will be trusted for documents (signed or certified) that chain to these Trusted Anchors. In the case of a CA certificate, you specify behavior that will be trusted for any signature that has a certificate issued by that CA. By configuring these activities you can, for example, distinguish if you will trust a certificate for signing or certifying.

A Trusted Anchor can be trusted for the following elements:

Trusted elements of Trusted Anchors

Trusted for	Description
Certified documents	Documents signed with this signature as an author signature, or whose certificate chain includes this certificate, are considered trusted for certified documents. Note: You must select this option if you want to select *Embedded High Privilege JavaScript*.
Embedded High Privilege Java Script	This option is only available when *Certified documents* is already selected. When enabled, JavaScript embedded in the document is allowed to be executed.
Signatures and as trusted root	Documents signed with this signature, or whose certificate chain includes this certificate, are considered trusted for signed documents. The certificate chain consists of the root certificate on the highest level and the dependent children certificates below. The Trusted Anchor of the Certificate Authority or entity can itself be a certificate used for digital signing and certifying. Do not choose this option if the Trusted Anchor is only expected to be in a signer's certificate chain. If you are certifying the document, you only need to select *Certified documents*; if the document must be signed and validated, you must choose this option.

If you install certificates, you should choose one or more of these options to specify what the certificate is trusted for. If you do not choose any options, the certificates are not trusted for any actions.

The table below shows which combinations of attributes for certificates are useful.

Useful combinations of attributes assigned to a certificate

Certified documents	Signatures and as trusted root	Description
X	-	Trust only children certificates for certifying.
-	X	Trust certificate itself and children certificates if the certificate is not issued by a CA. Trust children

		certificates for signing if public certificate is issued by a CA.
X	X	Trust certificate itself and children certificates for signing and certifying.

Procedure

To install a Trusted Anchor file:

1. Copy the Trusted Anchor file (`<filename>.cer`) to the */usr/sap/<SAPSID>/SYS/global/AdobeDocumentServices/ TrustManagerService/trust/certificates* directory.
2. Repeat these steps on each *Server* node. Note that these steps are not required on the *Dispatcher* node.

   💡

   If the *Server* nodes are running within a single cluster, the nodes are updated automatically and you do not have to repeat the steps.
3. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)
4. On the *Cluster* tab, choose *Server <x>* → *Services* → *Document Services Configuration*.
5. On the *Runtime* tab, select *Trusted Anchors*.
6. In the *Certificate File* field, choose *Browse* to search for the name of the Trusted Anchor file and then *Select*.
7. Select the actions that you want the Trusted Anchor to be *Trusted For*, and then click *Add*.
8. Restart the service *PDF Manipulation Module* for the changes to take effect. (See How to Restart a Service [Page 58].)

7.7.2 Installing Certificate Revocation Lists

Use

Certificate Revocation List files (CRLs) are provided by a CA, and identify credentials that can no longer be trusted. CRLs prevent you from applying a digital signature that is no longer valid, and let you know when digital signatures on incoming documents are invalid. CRLs should be updated on a regular basis (for example, daily or weekly).

They are identified by the CRL distribution point (CRLdp), which is specified as a URL in the certificate itself.

The following values must be specified when you install the CRL:

CRL Values

Value	Description
URL	Must match the URL found in the CRLdp field of the certificate.
Filename	The file name of the CRL.

⚠

If you do not specify a URL/file name combination, the server will not have access to CRLs so that signatures chaining off that Trusted Anchor are considered invalid. However, if the certificate does not contain a CRLdp field to identify a URL for its CRLs, revocation checking cannot be performed and the server will consider the signatures as always valid.

Procedure

To install a CRL file:

1. Copy the CRL file (`<filename>.crl`) to the */usr/sap/<SAPSID>/SYS/global/AdobeDocumentServices/TrustManagerService/trust/CRLs* directory.
2. Repeat this step on each *Server* node. Note that this step is not required on the *Dispatcher* node.

   💡

   If the *Server* nodes are running within a single cluster, the nodes are updated automatically and you do not have to repeat the steps.
3. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)
4. On the *Cluster* tab, choose *Server <x>* → *Services* → *Document Services Configuration*.
5. On the *Runtime* tab, select *CRLs*.
6. Specify the URL of the CRL you installed.
7. In the *Filename* field, choose *Browse* to search for the name of the CRL file, and then *Select*.
8. Click *Add*.
9. Restart the *PDF Manipulation Module* service for the changes to take effect. (See How to Restart a Service [Page 58].)

8 Licensing Adobe Document Services

Use

The Document Services License Service checks the license status of interactive form designs in your SAP system on a daily basis.

💡

Adobe LiveCycle Designer enables form authors to create new form designs or customize previously developed form designs. The form design provides the presentation or layout for the data, including formatting information such as font size, alignment, field logic, and graphics. The data from your SAP system populates the form design and determines what the final output will contain when Adobe document services processes the form design and data. The output can be either interactive forms or print forms.

The License Compliance indicator displays:

- A red light, which means that you require a license. See your SAP account representative to obtain a license to use SAP Interactive Forms.
- A green light, which means that your system is licensed. The green light appears in the case where you do not have a license and there are no customized forms, or where you are within your license agreement.

Forms are classified and displayed as:

- Customer Forms, the total number of your forms.
- SAP Forms, the total number of original SAP forms.
- Draft Forms, the total number of forms in a draft state. Draft forms are forms that are not currently in production and therefore do not count as a licensed/unlicensed form.

Prerequisite

The license for SAP Interactive Forms is an official SAP license. For further details, consult your contact person in your local SAP sales office. For more information, see SAP Note 750784.

Procedure

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)
2. On the *Cluster* tab, choose *Server <x>* → *Services* → *Document Services License Service*.

   The Document Services License Service displays the total number of SAP forms, customer forms, and draft forms processed to date.
3. Select *This server is licensed for the use of SAP Interactive Forms by Adobe*.

9 Adding Fonts

Use

Adobe document services require access to fonts that are installed with the Font Manager Module. This module contains a number of Adobe bundled fonts installed in `/usr/sap/<SAPSID>/JC<xx>/j2ee/os_libs/adssap/FontManagerService/fonts/adobe`.

You can also add fonts obtained from other vendors. The types of fonts you can add are OpenType® (.otf), TrueType® (.ttf), and PostScript® Type 1 (.pfb/.pfm).

Procedure

To add fonts:

1. Create a subdirectory called `fonts` below the `/usr/sap/<SAPSID>/SYS/global/AdobeDocumentServices/FontManagerService` directory.

   Enter **`JC<xx>`** if your system is a SAP Web AS J2EE system.

   Enter **`DVEBMGS<xx>`** if your system is a SAP Web AS ABAP + J2EE system (J2EE Add-In).

2. Create a subdirectory called `customer` below the `fonts/` directory created in the previous step.
3. Copy your fonts into the `/usrsap/<SAPSID>/SYS/global/AdobeDocumentServices/FontManagerService/fonts/customer` directory.
4. Restart the Document Services Font Manager for the changes to take effect. (See How to Restart a Service [Page 58].)
5. Also restart the application *com.adobe/AdobeDocumentServices* for the changes to take effect. (See How to Restart an Application [Page 59].)

10 Mapping Fonts

Use

Font mapping defines which fonts are used in a form if a specified font is not available. That is, you can specify which font to substitute for an unavailable font.

Font mapping information on the server is specified within XML tags in the Custom_XFA.xci file. You must specify the font mapping for whichever destination it applies to (for example, PDF, PCL, PostScript, or Zebra).

This sample code shows font mapping tags located in an XCI file.

```
<map>

<!--  [0..n] Attributes: from='', to='' [0..n]   -->

<equate from="Courier_*_*" to="Courier New_*_*" force="0"
/>

<equate from="CourierNew_*_*" to="Courier New_*_*"
force="0" />

<equate from="Courier-New_*_*" to="Courier New_*_*"
force="0" />

<equate from="MyriadPro_*_*" to="Myriad Pro_*_*" force="0"
/>
```

The map element contains font mapping information. Each font map is defined within an equate tag. The XML syntax for a font map is as follows:

> `<equate from="[`*original font_weight_posture*`]" to="[`*substitute font_weight_posture*`]"/>`

The font information for the original font and substitute font is defined as follows:

> *<Fontname_weight_posture>*

where:

- *Fontname* is the name of the font as it appears in the font list in LiveCycle Designer.
- *weight* determines whether the font is bold. Valid values are bold, normal, or * (wildcard).
- *posture* determines whether the font is italic. Valid values are italic, normal, or * (wildcard).

Procedure

To map fonts on the server:

1. Navigate to the `/<DIR_GLOBAL>/AdobeDocumentServices/lib` directory.
2. Map the fonts in the XCI file:
    - Modify the text file called Custom_XFA.xci.
    - If Custom_XFA.xci does not exist, copy the xfa.xci file in this directory to Custom_XFA.xci.

Example

The following XML code shows an example of an XCI file that contains font mapping information.

```
<zpl>

<fontInfo>

     <defaultTypeface writingScript="Roman">CG
     Triumvirate</defaultTypeface>

     <defaultTypeface writingScript="Japanese">Kozuka Mincho Pro-VI
     R</defaultTypeface>

     <defaultTypeface writingScript="SimplifiedChinese">Adobe Song
     Std L</defaultTypeface>

     <defaultTypeface writingScript="TraditionalChinese">Adobe Ming
     Std L</defaultTypeface>

     <defaultTypeface writingScript="Korean">Adobe Myungjo Std
     M</defaultTypeface>

<map>

     <equate from='MS Gothic_*_*' to='Kozuka Gothic Pro M_*_*'
     force="1"/>

</map>

</fontInfo>

</zpl>
```

11 Changing the Default PDF Version

Use

By default, SAP Interactive Forms renders output in PDF version 1.6. (supported by Reader versions 7.0 and higher) You may need to change this default setting if your forms have one of these features.

Features Requiring PDF 1.7 (supported by Reader versions 8.0 and higher)

- Digital signatures using seed values or filters – Signing options can be restricted to filter settings. Seed values are provided for the XFA processing to choose from.

- Saving a form with unbound data elements – With PDF 1.7, it is not necessary to use the custom object, Hidden Bound Text Field, which contains script for detecting changes in unbound form data for forms opened in Adobe Reader 7.0.5 or earlier.
- Digital signatures applied to a selected group of fields
- Hierarchical lists – The List and List Item subform roles allow you to create hierarchical lists. When a PDF form is generated, screen readers announce the list, the number of items, the nesting level, and the list end.
- Do Not Scroll option
- Comb fields – A comb field is a text field that is broken into separate segments for entering a set of numbers or characters.
- Transparent images – Transparent PNG and GIF files allow for overlaying images on forms.
- Performing validations before submit/print/save – If user-entered values do not match the validation pattern or LiveCycle Designer defaults, a programming error or warning appears.
- Automatic binding of list items in a Web Dynpro drop-down list – Previously, this connection was performed manually by the form designer.
- Different check mark shapes – Check marks can be populated by these shapes: check, circle, cross, diamond, square, star.
- Media selection – You can specify a particular size and type of paper for printing.
- Duplexing and complex page layout

Procedure

To change the default PDF version from 1.6:

1. Navigate to the `/<DIR_GLOBAL>/AdobeDocumentServices/lib` directory.
2. Open the Custom_XFA.xci file in a text editor and change the version setting. If Custom_XFA.xci does not exist, copy the xfa.xci file in this directory to Custom_XFA.xci:

```
<pdf> <!--  [0..n]  -->
      ...
      <version>1.7</version>
</pdf>
```

3. Save Custom_XFA.xci.

12 Managing XDC Files and Job Profiles

XDC Files

An XDC file is a printer description in XML format. Adobe document services require this file to create the print files. PDF-based forms can only be printed on printers whose SAP device type has an XDC file in the system.

The following XDC files are available:

- acrobat6.xdc – Supports data for rendering output in PDF 1.5 (retained for compatibility with previous versions of Adobe document services)
- acrobat7.xdc – Supports data for rendering output in PDF 1.6 (retained for compatibility with previous versions of Adobe document services)
- adobepdf.xdc – Supports data for rendering output in PDF 1.6 and 1.7
- hp4350pcl5e.xdc – For generating PCL output on an HP 4350 printer
- hp4350ps.xdc – For generating Postscript output on an HP 4350 printer
- hppcl5c.xdc – For use with PCL printers that support the HP PCL 5c printer language
- hppcl5e.xdc – For use with PCL printers that support the HP PCL 5e printer language
- lmt644pcl5e.xdc – For generating PCL output on a Lexmark T644 printer
- lmt644ps.xdc – For generating Postscript output on a Lexmark T644 printer
- ps_plain.xdc – For use with printers that support the PostScript printer language
- ps_plain_mt.xdc – For use with printers that support the PostScript printer language and use MT (Monotype) font names. For more information, see SAP Note 867662 and the documentation about the XDC scenarios, mentioned below.
- zpl203.xdc – For use with 203 dpi (8 dots/mm) Zebra label printers
- zpl300.xdc – For use with 300 dpi (12 dots/mm) Zebra label printers

When Adobe document services are deployed to the SAP NetWeaver AS, the XDC files are located in this directory: /<DIR_GLOBAL>/AdobeDocumentServices/lib.

In some cases, it may be necessary to make changes to these files. You upload XDC files to the server in the following cases:

- You want to install a new XDC file.
- You want to install a corrected XDC file.
- You want to install an XDC file that you have modified.

For more information on uploading and managing of the XDC files, see Administering XDC Files for SAP Device Types (Report RSPO0022) [Extern] and SAP Note 685571.

Job Profiles

Job profiles contain job-specific configuration settings in XML format. When your application calls Adobe document services, you can optionally specify a job profile. If you do, the profile is found and used to modify the way Adobe document services process that job.

When Adobe document services are deployed to the SAP NetWeaver AS, the job profiles are located in this directory: /<DIR_GLOBAL>/AdobeDocumentServices/JobProfiles/Adobe/Print

The following job profiles are shipped with Adobe document services:

- MediaLetterToLetterhead.xml – Maps the letter medium to letterhead.
- MediaA4ToA4Letterhead.xml – Maps the A4 medium to A4 letterhead.
- DuplexLongEdge.xml – Prints the document double-sided on the long edge
- DuplexShortEdge.xml – Prints the document double-sided on the short edge

- ZPLSettings.xml – Specifies label flipping, print mode, print speed, and media darkness (head temperature)

You can use the existing job profiles as examples to create customized settings for your print jobs.

XDC and Job Profile Scenarios

Scenarios provide you with examples of how you can make specific settings for your printer or print job. You can download the document *Customizing Printing Workflows* from the SAP Service Marketplace at **`http://service.sap.com/adobe`** → *Media Library* → *Documentation*.

13 Configuring GRMG Availability for the Adobe Document Services

Use

The Adobe Document Services are used to create and process PDF-based forms. These are used both in mass printing in backend systems and in interactive business processes.

You can use the GRMG to monitor the availability of the following components:

- Web Service interface
- XML Form module
- PDF Manipulation module

Prerequisites

A prerequisite for being able to activate the GRMG scenario is that the Web Dynpro PDF object is configured appropriately on the Adobe Document Services host. Make the exact settings, depending on the Web Service authentication procedure you are using, in accordance with the sections *Setting Up Basic Authentication in a Java Environment* or *Setting Up the SSL Connection in a Java Environment* of the document *Configuration Guide - Adobe Document Services*.

You can find the above document under the Quick Link *nw04installation* in the SAP Service Marketplace (http://service.sap.com/nw04installation).

Procedure

Perform the activation of the GRMG scenario as follows:

1. On the host of the J2EE Engine, switch to the following directory:

   Microsoft Windows: **`<J2EE Home directory>\JC<Inst. No.>\j2ee\admin\`**

   UNIX: **`<J2EE Home directory>/JC<Inst. No.>/j2ee/admin/`**

   If you installed the J2EE Engine together with an ABAP Engine, the character combination in the name of the **`JC<Inst. No.>`** directory in the above path may vary.

2. Start the Visual Administrator by calling the start script **go** (UNIX) or **go.bat** (Microsoft Windows), and log on with your user as the J2EE Engine administrator.

3. In the navigation bar, choose *Cluster* → *<SysID>* → *Server <x>* → *Services* → *Monitoring*, and choose the tab page *GRMG Customizing*.

4. Expand the left half of the screen. All applications that have a prepared GRMG Customizing file are displayed. Choose `sap.com/tc~ads~grmg`.

   The GRMG scenario is displayed in the Alert Monitor of the central monitoring system (CEN) in the subtree `GRMG:ADS <SysID>:<Host>`. To change the name of this subtree, select the entry *customizing → scenarios → scenario → scentexts → scentext → scendesc→ GRMG:ADS…* on the subscreen *Customizing tree for application*, choose the *Edit* button, and enter the desired name.

5. Transfer the Customizing file to the CCMS agent by choosing the *Upload* button.

Checking the Configuration (Optional)

6. To check whether the GRMG monitoring was successfully activated, call transaction GRMG in CEN, and choose *Upload/Download → Query CCMS Agent for Scenarios*.

   Without manual polling, you would have to wait up to an hour for the GRMG scenarios that you have uploaded in the Visual Administrator to be transferred to CEN and started.

7. The scenario that you have just activated is displayed with its URL (*scenstarturl*) and description (*scendesc*).

8. Call transaction RZ20 and start the monitor *Adobe Document Services* from the *SAP J2EE Monitor Templates* monitor set.

9. Expand the *Heartbeat* subtree. You can find availability information for the Adobe Document Services there.

Monitoring with the Generic Request and Message Generator [Extern] start page

14 Monitoring the Adobe Document Services EJB

You can view the following information on the Document Services EJB Monitor, using the Visual Administrator:

- Version numbers of XDC and XCI files
- Credential alias information
- Performance guidance information about credential status, and statistics about the number of EJB instances, transactions, and duration of transactions

Information about credential aliases is recovered from the Configuration Service and includes the expiry date and current status.

In addition to viewing information about the Document Services EJB, you can also configure the frequency that data is monitored, and the meaning of the colored performance indicators that display beside each resource.

For more information, see:

Viewing EJB Monitoring Information [Page 50]

Configuring Resource Monitoring Settings [Page 50]

14.1 Viewing EJB Monitoring Information

Procedure

To view EJB monitoring information:

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)
2. On the Cluster tab, choose *Server <x>* → *Services* → *Monitoring*.
3. Choose the *Monitor Tree* tab, and choose *Root* → *Services* → *Document Services EJB Monitor*.
4. Choose any of the following items to view information:
   - *Config Versions*: Lists each XDC or XCI file installed, and the version number for each file
   - *Credential Alias Entries*: Lists the alias, expiry date and status of each credential installed
   - *Credential Status*: States the current status of all credentials. The Green icon beside this item in the tree indicates that all credentials are valid
   - *Exceeded EJB Instances*: Displays number of EJB instances that exceed the XML Form Module PoolMax property. Use this information to determine if additional CPUs are required to handle all the EJB requests. For information about setting the PoolMax property, see Specifying the PoolMax Value [Page 57].
   - *Request Count*: Displays the total number of transactions. Choose *History* to view the transaction numbers according to various time intervals.
   - *Request Duration*: Displays the average duration of each EJB request since the server startup
5. Configure the way that the monitoring information is reported. See Configuring Resource Monitoring Settings [Page 50].

14.2 Configuring Resource Monitoring Settings

Use

The colored icons beside *Credential Status*, *Exceeded EJB Instances*, *Request Count* and *Request Duration* provide an indication of the performance level of each resource when it was last monitored. You can specify how the performance levels are determined, and also how often the monitoring service polls the resource monitor for new data.

Prerequisites

To configure monitoring frequency and performance indicators:

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)
2. On the *Cluster* tab, choose *Server <x>* → *Services* → *Monitoring*.
3. Choose the *Monitor Tree* tab, and choose *Root* → *Services* → *Document Services EJB Monitor*.
4. Choose any of the following items:

- *Credential Status*
- *Exceeded EJB Instances*
- *Request Count*
- *Request Duration*

5. On the *General* tab of the *Monitor Configuration* dialog box that appears, choose *Performance*, and choose *Edit*.
6. In the *Data Collection* area, under the *Polled by Monitor* option, set the frequency that the monitoring service polls the resource monitor for new data:
   - *Number*: Enter the number of times per unit that the resource monitor is polled.
   - *Units*: Choose the unit of time measurement (minutes, hours or days) that determines how often the resource monitor is polled.
7. Under *React on resource failure*, choose the action that the server takes if the monitoring service fails to obtain data from the resource due to an exception:
   - *Ignore*: The server ignores the failure.
   - *Unregister monitor*: The server unregisters the monitored resource.
8. Choose the *Performance* tab, and choose *Edit*.

   The Credential Status resource has a *State* tab instead of a *Performance* tab. The *State* tab displays the characteristics of each flag color: Green indicates that the credential is valid, yellow indicates a warning that the credential will expire soon, and red indicates that the credential has expired.
9. For each of the fields on the dialog box, type the number at which you want the performance flag to change to the next color indication.

   You would like to set the performance indicator for Exceeded EJB Instances. If the pool max value is 25, you could set the flag to change from green (acceptable) to yellow (caution) when the number of server instances reaches 10.
10. Choose *Save*, and repeat steps 2 to 9 for each monitored resource you want to configure.

15 Additional Installations on the Client PC

To use SAP Interactive Forms on a client PC, you need to install the following components on the client:

- Adobe Reader or Adobe Acrobat (Version 7.0.9 or higher)

  You need Adobe Reader or Acrobat to preview SAP Interactive Forms at design time or to display them in your Web browser. You may need to do this, for example, when you edit fields in online or offline scenarios.

  You can have both Adobe Reader and Acrobat installed on the client. You can choose whether Adobe Reader or Acrobat is used to open a PDF document. You can configure this during the installation procedure or in the configuration settings of your system. In a Microsoft Windows environment, for example, choose *Start* → *Settings* → *Control Panel*. Then choose *Folder Options* → *File Types* to specify which program you want to use to open PDF files.

For additional and current information about the required version of Adobe Reader or Acrobat, refer to SAP Note 834573. You can obtain a free download of the Adobe Reader from the Adobe website at http://www.adobe.com/products/acrobat/readstep2.html.

- SAP Web Dynpro Active Component Framework (ACF)

  The ACF is a framework for integrating Active Components such as ActiveX and Java Applets into Web Dynpro. To install ACF, you need to have administrator authorization on the client PC. To run ACF, you must enable ActiveX in your Web browser. For additional and current information about installing the ACF, refer to SAP Note 766191.

In releases as of SAP NetWeaver 7.0 SPS10 you can use interactive forms in the Web Dynpro runtime environment based on Zero Client Installation (ZCI) technology. ZCI enables you to use interactive forms in Adobe Reader without any additional plug-ins, and also has the benefit of being platform-independent. This means that ACF is no longer required for interactive forms. Forms for ZCI, however, must include special scripting. You must migrate any interactive forms that are not yet ZCI-enabled. For more information, see SAP Notes 956074, 947675 and 1042394.

16 Running Adobe Document Services

Adobe document services consist of the Adobe document services Web service and these SAP J2EE services:

- Document Services Data Manager
- Document Services Font Manager
- Document Services License Service
- Document Services Trust Manager Service
- PDF Manipulation Module (High Encryption or Low Encryption)
- XML Form Module

These services are installed together with SAP NetWeaver AS Java and must all be running to enable Adobe document services to operate correctly. In the event of problems, you can verify the state of these services using the Visual Administrator.

These services are exclusively for Adobe document services and must not be used for any other applications.

16.1 Problem Analysis

If errors with, for example, connections or the configuration occur when you work with Adobe document services, you can find documents in the SAP Library that contain Problem Analysis Scenarios. These documents assist you with troubleshooting. For more information, see the following links in the SAP Library:

- Problem Analysis Scenarios for Adobe Document Services [Extern] in the *SAP NetWeaver Library* → SAP *NetWeaver by Key Capability* → *Solution Life Cycle Management by Key Capability* → *SAP NetWeaver™ Problem Analysis Guide (PAG)* → *SAP Web Application Server Problem Analysis Scenarios*

- When you work with PDF-Based Print Forms in high-volume printing scenarios using the ABAP Workbench, you can also use the function Saving Runtime Information and a Generated PDF Locally [Extern] to investigate any issues. See *SAP NetWeaver Library → SAP NetWeaver by Key Capability → Application Platform by Key Capability → Business Services SAP → Interactive Forms by Adobe → PDF-Based Print Forms → Calling Forms in an Application Program*.

16.2 Changing the Design of the ERROR.PDF File

Use

When Adobe document services render a form, there may be problems with the rendering or there may be problems using the form after rendering. The problem analysis scenario Adobe Rendering Error [Extern] helps you to examine the reason for these problems. The ERROR.PDF file provided uses a default layout, which you can change for your own requirements.

Prerequisites

- SAP NetWeaver 7.0 SP Stack 6 or higher.
- Adobe LiveCycle Designer is installed on your front end.
- You need to have access to the server that hosts the Adobe document services.

Procedure

1. Start the Adobe LiveCycle Designer in stand alone mode.
2. Open the file `LOG.PDF` in Adobe LiveCycle Designer. This file is stored on the server that hosts the Adobe document services at `<DIR_GLOBAL>\AdobeDocumentServices\lib`.
3. Make your changes in the layout. For example, you can insert your own logo.

   ⚠

   Do not make any changes to the schema that is assigned to the layout.
4. Rename the file as `CUSTOM_LOG.XDP`.
5. Save the file as Dynamic PDF.
6. Exit Adobe LiveCycle Designer.

Result

You have customized the design of the ERROR.PDF that is used for displaying the Adobe rendering error.

16.3 Changing the Maximum Size for the Storage of the ERROR.PDF File

Use

When the rendering of a form fails, Adobe document services create a file that contains detailed information about the error, called Error.PDF. This file is written on the server that hosts the Adobe document services in the directory `<DIR_GLOBAL>\AdobeDocumentServices\renderErrorLog\ErrorFiles`. The file name is `<Date+Time+ApplicationName+Username>.pdf.` After it creates the

Error.PDF, the system examines the directory to determine the total size of files in the directory. If that size is more than the maximum allowed, it begins deleting files (oldest first) until the directory size is below the maximum allowed or until only one error file is left in the directory.

> ⚠
>
> The default size of the error file directory is 0 MB, because if your forms contain security-related data or information, you may avoid storing Error.PDF files. To enable the storage of Error.PDF files you have to set the value for the <threshold>.
>
> Note that this directory does not exist initially. The system creates it automatically the first time a rendering error occurs.

Prerequisites

SAP NetWeaver 7.0 SP Stack 6 or higher

Procedure

If you want to change the maximum size, do the following:

1. Change the value by modifying the file `renderErrorConfig.xml` in the directory `<DIR_GLOBAL>\AdobeDocumentServices\renderErrorLog.`
2. Save the file
3. Start the J2EE Engine for the change to take effect.

Example

Here is a sample `renderErrorConfig.xml:`

```
<?xml version=“1.0“?>
<renderErrorLog>
   <!–The maximum size of error files directory, measure in MB-->
   <threshold>
      100
   </threshold>
</renderErrorLog>
```

See also:

Adobe Rendering Error [Extern]

16.4 Viewing the Logs

Use

All of the components of Adobe document services work together and record events to the logs, including any service errors. Security-related messages are logged in the security.log file of the J2EE Engine. This log file contains information on the user and the actions he performed like certifying or signing a form and the used credentials. An easy way to find the appropriate information is to search for the location com.adobe.AdobeDocumentServices. This filters out entries that match the following pattern:

Date, time: User: <name> ……..action that was performed ……..with credential alias <alias>

Wed Jul 20 11:07:19 CEST 2005 User: SCHMIDT successfully signed a PDF from <InputPDF> Source: Stream Name: PDF with credential alias cert_credential

Procedure

To access the logs:

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)
2. On the *Cluster* tab, choose *Server <x>* → *Services* → *LogViewer*.
3. On the *Runtime* tab in the right panel, choose *Server <x>* → *<..>\usr\sap* → *<SAPSID\JC<xx>* → *j2ee\cluster\server<x>*.
4. Below the system node, there are logs in which you can view any errors and fatal messages logged by Adobe document services:
   - `server.log`
   - `security.log`
   - `database.log`

   Select the appropriate log entry to open the log.
5. Choose *defaultTrace.trc* to view any trace log entries for Adobe document services. For more information, see Activating the Trace for Adobe Document Services [Page 55].

16.5 Activating the Trace for Adobe Document Services

Use

To analyze a problem with Adobe document services, switch on the trace and reproduce the problem. The trace entries then provide you with detailed control flow information.

If you activate the trace for Adobe documents services, all business data to be printed out or displayed in a PDF, will be recorded as Base 64-encoded data in the trace file. To avoid the storage of productive or security-related data, do not activate the trace.

Prerequisites

You need to have Administrator user rights.

Procedure

1. Start the Visual Administrator and logon as the Administrator user. (See How to Start the Visual Administrator [Page 57])
2. Expand *Server* → *Services* in the tree on the left and choose the *Log Configurator* node.
3. In the new panel on the right side, choose the *Locations* tab.
4. Expand the *com* node and choose *adobe* in the *Log Controllers* tree.
5. In the panel on the right side, set the value of *Severity* to *All* and click the *Apply* button.

Result

The trace entries of the Adobe document services runtime are written to the default trace file defaultTrace.trc which is located in the log directory of the server node for example, C:\usr\sap\J2E\JC00\j2ee\cluster\server0\log on a Windows machine.

Don't forget to set the *Severity* back to value *Error* after the requested trace entries are written to the trace file.

17 Configuring Multi-Processing

The overall performance of the Adobe document services can be improved using parallel processing. This significantly enhances the throughput of, for example rendered documents. Ensure that the following two prerequisites for parallel processing are fulfilled:

Prerequisites

- The application sends requests in parallel (for example, multiple background payroll jobs for different groups of employees).
- The Adobe document services run on multiprocessor hardware.

Services

Adobe document services control the number of simultaneous requests that can be processed at any given time. The value is controlled by the PoolMax setting in each of the services. As with all performance tuning, establishing the right values for these services encompasses balancing all of the applications running on the server.

Adobe document services provide two services for performing specific functions:

- Print output renders the form to a printer language and/or PDF. No further manipulation of the PDF is normally done. The XML Form Module is used to render a form.
- The PDF Manipulation Module is used to retrieve data, metadata, and other information from a PDF file.

Producing interactive forms for e-mail or for online use usually involves both the XML Form Module and the PDF Manipulation Module.

The memory requirements for each process of the Adobe document services are as follows:

- XML Form Module: 25 MB approximately
- PDF Manipulation Module: 30 MB approximately

These services handle separate processes at the operating system level (for example, XMLForm.exe and PDFManipulation.exe on Microsoft Windows). You use the Visual Administrator to adjust the maximum number of allowed processes for each of these services by setting the PoolMax attribute. The default values are:

PoolMax Default Values

Service	PoolMax Value
XML Form Module	4

PDF Manipulation Module	2

We recommend that you set the PoolMax value of the XML Form Module to the maximum numbers of processors used by your application.

Example

The following table shows the performance improvements that can be achieved on a server with four processors:

Multi-Processor Performance Differences

PoolMax Value of XML Form Module	Increase of throughput
1	100% (base value)
2	190%
3	270%
4	345% (optimal value)

These settings can also be used to minimize the impact of the document services on other applications that are installed on the same server. For example, if you set the PoolMax value of the XML Form Module to 2, a maximum of two processors will be used by XMLForm.exe processes. Then other software could run simultaneously if there are additional processors.

17.1 Specifying the PoolMax Value

Procedure

To specify the PoolMax value for each module, do the following:

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)
2. Choose the service you want to specify the PoolMax Value for: On the *Cluster* tab, choose *Server <x>* → *Services* →
   - ○ *PDF Manipulation Module* – (*High Encryption* or *Low Encryption*)
   - ○ *XML Form Module*
3. For each service mentioned in the previous step, on the *Properties* tab in the right panel, enter the value in the *Value* box next to the *PoolMax Key*.

18 How to Start the Visual Administrator

Procedure

1. Start the tool:

- For an ABAP + J2EE system (J2EE Add-In):
  - On Windows:

    Run **`\usr\sap\<SAPSID>\DVEBMGS<xx>\j2ee\admin\go.bat`**.
  - On UNIX:

    Run **`/usr/sap/<SAPSID>/DVEBMGS<xx>/j2ee/admin/go`**.
- For a J2EE system:
  - On Windows:

    Run **`\usr\sap\<SAPSID>\JC<xx>\j2ee\admin\go.bat`**.
  - On UNIX:

    Run **`/usr/sap/<SAPSID>/JC<xx>/j2ee/admin/go`**.

The *SAP J2EE Engine – Administration* screen with the *Connect to SAP J2EE Engine* dialog box appears.

2. To connect, do the following:

Connecting to the SAP J2EE Engine

J2EE type	How to connect
J2EE system	Choose *Connect* to use the Default login and enter the password for the Administrator user of the SAP J2EE engine.
J2EE Add-In	You cannot use the Default login. Instead do the following: 1. Choose *New*. 2. Enter a display name and choose *Direct Connection to a dispatcher Node*. 3. Choose *Next*. 4. Enter at least the following: ○ *User Name*: **`J2EE_ADMIN`** ○ *Host*: **`<host_name>`** of the J2EE engine ○ *Port*: **`<P4_Port>`** The following convention applies for the port: ***5<J2EEinstance_number>04***. For example, if your J2EE instance number is **15**, the P4port is 5**15**04. 5. Choose *Save* and connect with your new login account by choosing *Connect*. 6. Enter the password for the J2EE_ADMIN user and choose *Connect*.

18.1 How to Restart a Service

Procedure

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)
2. On the *Cluster* tab, choose *Server <x>* → *Services* → *<service to start/stop>*.

3. For stopping the service, choose *Stop service*.
4. For restarting the service, choose *Start service*.

18.2 How to Restart an Application

Procedure

1. Log on to the Visual Administrator. (See How to Start the Visual Administrator [Page 57].)
2. Choose *Server <x>* → *Services* → *Deploy*.
3. Choose *Application*.
4. Choose the application you want to restart in the tree.
5. Choose *Stop Application*.
6. For restarting the application, choose *Start Application*.